EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS
ISSN 2306–6172
Volume 12, Issue 1 (2024) 57 – 69

A BLIND COLOR WATERMARKING TECHNIQUE BASED ON QUATERNION COMPLEX STRUCTURE-PRESERVING QR DECOMPOSITION

Guo Z.

Abstract The color image blind watermarking technique is widely used in the field of digital image processing and security and can maintain the copyright of images and guarantee their authenticity. In order to maintain the intrinsic connection between the three color channels of color images, using quaternion algebra to solve the color image problems often achieves better results. This paper proposes a complex structure-preserving algorithm for solving the QR decomposition of quaternion matrices, which has better computational efficiency compared with previous methods. This paper also applies the proposed algorithm to the color image blind watermarking technique, and experiments prove that the method has better invisibility and better robustness to salt and pepper noise, etc. The technique will have potential applications in the field of color image watermarking.

Key words: Quaternion matrix; Complex structure-preserving algorithm; QR decomposition; 6-bit trinary; Blind color watermarking technique.

AMS Mathematics Subject Classification: 15B33, 68U10.

DOI: 10.32523/2306-6172-2024-12-1-57-69

1 Introduction

In recent years, color image watermarking techniques have received extensive attention from researchers and play important and indispensable roles in related application fields [1-6]. Because color watermarking techniques can embed more watermark information, and have better robustness and invisibility than the traditional grayscale watermarking algorithms [7]. However, the classical color watermarking algorithms mainly include single-channel processing, multi-channel synthesis, etc., which essentially operate on grayscale images and do not take into account the intrinsic connection between the three color channels of red, green, and blue.

It was not until the introduction of quaternions that conditions were created for a reasonable explanation of this problem [8]. In 1997, S. Pei [9] first introduced that a pure imaginary quaternion model could be used to represent color images. Since then, color image processing algorithms based on quaternion algebra have been developed rapidly and often achieve unexpectedly good experimental results. In 2003, P. Bas [10] proposed a digital watermarking scheme for color images based on the quaternion Fourier transform. In 2016, X. Wang [11] presented a color watermarking method for the local quaternion polar harmonic transform, which has better invisibility and robustness for adding distorted images such as median filtering. In 2021, Y.

Chen [12] introduced a blind watermarking algorithm by means of the real structure-preserving quaternion QR decomposition, which can be more resistant to attack for brightness adjustment and Poisson noise while reducing the computational complexity. In 2023, M. Zhang [13] introduced a color watermarking technique based on the complex structure-preserving quaternion singular value decomposition algorithm, and proved the effectiveness of the algorithm through experiments.

In this paper, we propose a new quaternion matrix QR decomposition method based on complex structure-preserving algorithm, which has higher computational efficiency compared with existing algorithms. Moreover, we also apply the proposed method to the blind watermarking problem of color images, and the experiments prove that the proposed method is effective. The blind watermarking algorithm is a method that does not require the original image as a reference during the watermark extraction. Most of the color watermarking algorithms based on QR decomposition need to segment the host image into many 4×4 or larger blocks and convert the decimal watermark image into a 8-bit binary array. The difference of this algorithm is to partition the host image into many small 3×3 blocks and to convert the decimal watermark image into a 6-bit trinary array using a quaternion model considering the intrinsic connection between the three channels of the color image. This method can effectively improve the embedding capacity of conventional watermarking algorithms.

The novelty of this paper consists of three main points:

1) In this paper, a new algorithm (complex structure-preserving algorithm) is proposed for solving the quaternion matrix QR decomposition problem.

2) Compared to the existing algorithms, the algorithm achieves an advantage in computational speed.

3) The blind color watermarking technique based on the quaternion matrix complex structure-preserving algorithm proposed in this paper has better performance.

The arrangement of this paper is as follows. In Second 2, we introduce some symbolic representations and concepts that need to be used in this paper. In Section 3, we explain the complex structure-preserving algorithm for QR decomposition of quaternion matrices (CQQRD) and test its performance. In Section 4, we presented the blind watermarking algorithm for color images based on CQQRD. In Section 5, we test the proposed color blind watermarking algorithm by numerical experiments and compare it with two other watermarking algorithms to verify the effectiveness of the method in this paper.

2 Preliminaries

2.1 Quaternion and its complex representation matrix

For arbitrary quaternion $q = q_1 + q_2\mathrm{i} + q_3\mathrm{j} + q_4\mathrm{k} \in \mathbf{H}$, its imaginary units i, j and k satisfy the following relations,

$$\mathrm{i}^2 = \mathrm{j}^2 = \mathrm{k}^2 = \mathrm{ijk} = -1, \mathrm{ij} = -\mathrm{ji} = \mathrm{k}. \tag{2.1}$$

The conjugate and the norm of q are defined as follows,

$$\overline{q} = q_1 - q_2\mathrm{i} - q_3\mathrm{j} - q_4\mathrm{k},\ \|q\| = \sqrt{q\overline{q}} = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2}. \tag{2.2}$$

Similarly, for any quaternion matrix $A = A_1 + A_2\mathrm{i} + A_3\mathrm{j} + A_4\mathrm{k} \in \mathbf{H}^{m\times n}$, then the conjugate transpose matrix and the Frobenius norm of A are shown below,

$$A^H = A_1^T - A_2^T\mathrm{i} - A_3^T\mathrm{j} - A_4^T\mathrm{k},\ \|A\|_F = \sqrt{\|A_1\|_F^2 + \|A_2\|_F^2 + \|A_3\|_F^2 + \|A_4\|_F^2}. \quad (2.3)$$

Definition 2.1 *For any quaternion matrix* $A = A_1 + A_2\mathrm{i} + A_3\mathrm{j} + A_4\mathrm{k} = A_\alpha + A_\beta\mathrm{j} \in \mathbf{H}^{m\times n}$*, where* $A_1, A_2, A_3, A_4 \in \mathbf{R}^{m\times n}, A_\alpha = A_1 + A_2\mathrm{i}, A_\beta = A_3 + A_4\mathrm{i} \in \mathbf{C}^{m\times n}$*, then the complex representation matrix* A^σ *of* A *was defined to be [14]*

$$A^\sigma = \begin{bmatrix} A_\alpha & A_\beta \\ -\overline{A_\beta} & \overline{A_\alpha} \end{bmatrix} \in \mathbf{C}^{2m\times 2n}. \quad (2.4)$$

Given three quaternion matrices $A \in \mathbf{H}^{m\times n}, B \in \mathbf{H}^{n\times p}$, then

$$(AB)^\sigma = A^\sigma B^\sigma, \quad (AB)_c^\sigma = A^\sigma B_c^\sigma, \quad (AB)_r^\sigma = A_r^\sigma B^\sigma, \quad (A^H)^\sigma = (A^\sigma)^H, \quad (2.5)$$

where the subscript $_{c,\,r}$ and the superscript ^{H} indicate the first column block, the first row block and the conjugate transpose of the related matrices, respectively. It is easy to prove that $A \to A^\sigma$ is an isomorphic mapping of $\mathbf{H}^{m\times n}$ on $\mathbf{C}^{2m\times 2n}$ by (2.4-2.5). Moreover, the quaternion matrix A is a unitary quaternion matrix if and only if A^σ is a unitary matrix.

2.2 Color image representation based on quaternion matrix

S. Pei [9] first proposed a model to represent color images with the help of a pure imaginary quaternion matrix in 1997. Therefore, the specific form of an $m \times n$ color image A can be shown as follows,

$$A = \mathbb{R}\mathrm{i} + \mathbb{G}\mathrm{j} + \mathbb{B}\mathrm{k} \in \mathbf{H}^{m\times n}, \quad (2.6)$$

in which $\mathbb{R}, \mathbb{G}, \mathbb{B} \in \mathbf{R}^{m\times n}$ denote the three color channels of A: red, green and blue, respectively.

Peak Signal-to-Noise Ratio (PSNR) is an important measure of image quality that can be used to evaluate the difference between the original image and a distorted or compressed image. The PSNR is usually measured in decibels (dB). Therefore, the PSNR value of two color images can be calculated by the following formula,

$$\mathrm{PSNR} = 10 \times \lg \frac{3mn \times 255^2}{\sum_{x=1}^m \sum_{y=1}^n \sum_{z=1}^3 (A(x,y,z) - \mathbb{A}(x,y,z))^2}, \quad (2.7)$$

where $A(x,y,z)$ and $\mathbb{A}(x,y,z)$ denote the pixel values at coordinate positions (x,y,z) of the original image and the distorted image, respectively. Generally, if the value of PSNR is greater than 30, then the distorted image has a good quality.

Normalized Correlation (NC) is a metric used to measure the similarity between two images. Usually in the watermarking algorithm, use the NC value to determine the relationship between the extracted watermark image and the original watermark image, and the NC calculation formula is defined as follows.

$$\mathrm{NC} = \frac{\sum_{x=1}^m \sum_{y=1}^n \sum_{z=1}^3 W(x,y,z) \times \mathbb{W}(x,y,z)}{\sqrt{\sum_{x=1}^m \sum_{y=1}^n \sum_{z=1}^3 W(x,y,z)^2}\sqrt{\sum_{x=1}^m \sum_{y=1}^n \sum_{z=1}^3 \mathbb{W}(x,y,z)^2}}, \quad (2.8)$$

where $W(x,y,z)$ and $\mathbb{W}(x,y,z)$ denote the pixel values at coordinate positions (x,y,z) of the original watermark image and the extracted watermark image, respectively. Generally, if the value of NC is closer to 1, it means that the two watermark images are more similar.

2.3 Scrambling and random position embedding methods for color images

Arnold scrambling is an image encryption and processing technique based on permutation operations, which can efficiently scrambled images and is widely used in the field of image protection and processing. Its theoretical basis is shown below [15],

$$\begin{bmatrix}\phi_1\\ \varphi_1\end{bmatrix} = \begin{bmatrix}a & b\\ c & d\end{bmatrix}\begin{bmatrix}\phi\\ \varphi\end{bmatrix}\mathrm{mod}(M), \qquad \begin{bmatrix}\phi\\ \varphi\end{bmatrix} = \begin{bmatrix}a & b\\ c & d\end{bmatrix}^{-1}\begin{bmatrix}\phi_1\\ \varphi_1\end{bmatrix}\mathrm{mod}(M), \tag{2.9}$$

where $(\phi,\varphi),(\phi_1,\varphi_1)$ denote the coordinate positions before and after image scrambling, respectively, and a,b,c,d are all integers and satisfy $ad-bc=1$. Moreover, mod() is the modulo operation, M is the dimension of the square image. Therefore, with the different selection of a,b,c,d, then we will get the different scrambled images, this is a modern common image encryption method. To ensure uniformity and convenience, we use $a=b=c=1, d=2$ as demonstration for Arnold scrambling in the paper.

Generating randomly embedded coordinate sequences. In order to increase the security and robustness of digital images, the method of choosing random position embedding is widely used, which can make it difficult for the attacker to confirm the specific position of embedding and thus make it more difficult for the attacker to crack the digital watermark. Random position embedding also prevents the watermark information from being too dense in a certain region and thus affecting the invisibility of the host image. In addition, it effectively resists common attacks such as clipping, scaling, rotation, etc., thus enhancing the robustness of the digital watermarking algorithm. The random position embedding method used in this paper is shown below [12].

Step 1. Determine the number of blocks N^2 that the host image can be divided into by the floor function, where $N=\mathrm{floor}(M/d)$, M and d denote the host image and the order of each block to be divided, respectively.

Step 2. Construct the sequence matrix $S\in\mathbf{R}^{2\times N^2}$,

$$S = \begin{bmatrix}\mathrm{kron}(1:N, \mathrm{ones}(1,N))\\ \mathrm{kron}(\mathrm{ones}(1,N), 1:N)\end{bmatrix}, \tag{2.10}$$

where kron() is the Kronecker tensor product, and $\mathrm{ones}(1,N)$ is a $1\times N$ all-1 vector. Then, it is easy to find that the transpose $S(:,t)^T$ of each column of S represents the row and column coordinates of the un-randomized matrix blocks, $t\in\{1,2,\cdots,N^2\}$.

Step 3. Generate a new random sequence of coordinates S_1 by the function randperm().

$$S_1 = S(:,\mathrm{randperm}(N^2))\in\mathbf{R}^{2\times N^2}. \tag{2.11}$$

3 Complex structure-preserving method for QR decomposition of quaternion matrices

QR decomposition of matrices, as an important part of matrix computation, plays an important role in problems such as eigenvalue computation of matrices, solution of linear systems of equations, vector orthogonalization, etc., and is popular in the field of signal and image processing. In this section, we will propose a complex structure-preserving QR decomposition algorithm for quaternion matrix based on complex Householder transformation and complex Givens transformation, which can effectively reduce the time spent on the algorithm.

Lemma 3.1. *[16] For an arbitrary quaternion matrix $A \in \mathbf{H}^{m\times n}$, then there exists a unitary quaternion matrix $Q \in \mathbf{H}^{m\times m}$ and an upper triangular quaternion matrix $R \in \mathbf{H}^{m\times n}$ such that $A = QR$.*

Here we take the first column block of the complex representation of a 5×4 quaternion matrix as an example to show the steps of QR decomposition of the complex structure-preserving algorithm based on generalized Householder transform H_t and generalized Givens transform G_t, in which $t = 1, 2, 3, 4$, and their definitions are given later in the algorithm. Then we can get an upper triangular quaternion matrix with real diagonals.

$$
\left[\begin{array}{cccc}
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times \\ \hline
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times \\
\times & \times & \times & \times
\end{array}\right]
\xRightarrow{H_1}
\left[\begin{array}{cccc}
a & a & a & a \\
0 & a & a & a \\
0 & a & a & a \\
0 & a & a & a \\
0 & a & a & a \\ \hline
a & a & a & a \\
0 & a & a & a \\
0 & a & a & a \\
0 & a & a & a \\
0 & a & a & a
\end{array}\right]
\xRightarrow{G_1}
\left[\begin{array}{cccc}
\tilde{a} & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & a & a & a \\
0 & a & a & a \\
0 & a & a & a \\
0 & a & a & a \\ \hline
0 & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & a & a & a \\
0 & a & a & a \\
0 & a & a & a \\
0 & a & a & a
\end{array}\right]
\xRightarrow{H_2}
\left[\begin{array}{cccc}
\tilde{a} & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & b & b & b \\
0 & 0 & b & b \\
0 & 0 & b & b \\
0 & 0 & b & b \\ \hline
0 & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & b & b & b \\
0 & 0 & b & b \\
0 & 0 & b & b \\
0 & 0 & b & b
\end{array}\right]
\xRightarrow{G_2}
\left[\begin{array}{cccc}
\tilde{a} & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & \tilde{b} & \tilde{b} & \tilde{b} \\
0 & 0 & b & b \\
0 & 0 & b & b \\
0 & 0 & b & b \\ \hline
0 & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & 0 & \tilde{b} & \tilde{b} \\
0 & 0 & b & b \\
0 & 0 & b & b \\
0 & 0 & b & b
\end{array}\right]
$$

$$
\xRightarrow{H_3}
\left[\begin{array}{cccc}
\tilde{a} & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & \tilde{b} & \tilde{b} & \tilde{b} \\
0 & 0 & c & c \\
0 & 0 & 0 & c \\
0 & 0 & 0 & c \\ \hline
0 & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & 0 & \tilde{b} & \tilde{b} \\
0 & 0 & c & c \\
0 & 0 & 0 & c \\
0 & 0 & 0 & c
\end{array}\right]
\xRightarrow{G_3}
\left[\begin{array}{cccc}
\tilde{a} & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & \tilde{b} & \tilde{b} & \tilde{b} \\
0 & 0 & \tilde{c} & \tilde{c} \\
0 & 0 & 0 & c \\
0 & 0 & 0 & c \\ \hline
0 & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & 0 & \tilde{b} & \tilde{b} \\
0 & 0 & 0 & \tilde{c} \\
0 & 0 & 0 & c \\
0 & 0 & 0 & c
\end{array}\right]
\xRightarrow{H_4}
\left[\begin{array}{cccc}
\tilde{a} & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & \tilde{b} & \tilde{b} & \tilde{b} \\
0 & 0 & \tilde{c} & \tilde{c} \\
0 & 0 & 0 & d \\
0 & 0 & 0 & 0 \\ \hline
0 & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & 0 & \tilde{b} & \tilde{b} \\
0 & 0 & 0 & \tilde{c} \\
0 & 0 & 0 & d \\
0 & 0 & 0 & 0
\end{array}\right]
\xRightarrow{G_4}
\left[\begin{array}{cccc}
\tilde{a} & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & \tilde{b} & \tilde{b} & \tilde{b} \\
0 & 0 & \tilde{c} & \tilde{c} \\
0 & 0 & 0 & \tilde{d} \\
0 & 0 & 0 & 0 \\ \hline
0 & \tilde{a} & \tilde{a} & \tilde{a} \\
0 & 0 & \tilde{b} & \tilde{b} \\
0 & 0 & 0 & \tilde{c} \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{array}\right]
$$

Next, we give the steps of the corresponding complex structure-preserving QR decomposition algorithm for quaternion matrices (CQQRD).

Complex structure-preserving method for QR decomposition of a quaternion matrix.
Given an arbitrary quaternion matrix $A_1 + A_2\mathrm{i} + A_3\mathrm{j} + A_4\mathrm{k} \in \mathbf{H}^{m\times n}$, without loss of generality, let $m \geq n$, find a unitary quaternion matrix $Q \in \mathbf{H}^{m\times m}$ and an upper triangular quaternion matrix $R \in \mathbf{H}^{m\times n}$ such that $A = QR$.
Input: the first column block A_c^σ of A^σ. Output: Q^σ and the first column block R_c^σ of R^σ.

Function: $[Q^\sigma, R_c^\sigma]$ =QC_QR(A_c^σ)
1. Initialization
   $[m_1, n] = \mathrm{size}(A_c^\sigma)$; $B = A_c^\sigma$; $m = m_1/2$; $P = [\mathrm{eye}(m); \mathrm{zeros}(m, m)]$;
2. for $s = 1 : n$
3. Determine if the elements below the diagonal of each column are zero.
   if $\mathrm{norm}([B(s+1:m, s); B((m+s+1):(2*m), s)]) > 0$
4. Perform the Householder transformation on the matrices B and P.
   $[u, \beta]$ =QC_House($B(s:m, s), -\mathrm{conj}(B(m+s:2*m, s)), m-s+1$);
   $B([s:m, s+m:2*m], s:n) = B([s:m, s+m:2*m], s:n) - (\beta * u)*$
   $(u' * B([s:m, s+m:2*m], s:n))$;
   $P([s:m, s+m:2*m], 1:m) = P([s:m, s+m:2*m], 1:m) - (\beta * u)*$
   $(u' * P([s:m, s+m:2*m], 1:m))$;
   end
5. Perform the Givens transformation on the diagonal elements of matrices B and P.
   G =QC_Givens($B(s, s), -\mathrm{conj}(B(s+m, s))$);
   $B([s, s+m], s:n) = G' * B([s, s+m], s:n)$;
   $P([s, s+m], 1:m) = G' * P([s, s+m], 1:m)$;
   end
6. Output results.
   Q^σ =Q_Comp($P(1:m,:), -\mathrm{conj}(P(m+1:2*m,:)))'$; $R_c^\sigma = B$;

Define the three subfunctions Q_Comp, QC_House and QC_Givens required above.

Function: x^σ=Q_Comp($x_1 + x_2\mathrm{i}, x_3 + x_4\mathrm{i}$)

$$x^\sigma = \begin{bmatrix} x_1 + x_2\mathrm{i} & x_3 + x_4\mathrm{i} \\ -x_3 + x_4\mathrm{i} & x_1 - x_2\mathrm{i} \end{bmatrix};$$

end

Function: $[u, \beta]$=QC_House(x_1, x_2, n)
   $u_1(1:n, 1:2) = [x_1, x_2]$; $a = \mathrm{norm}([x_1; x_2])$; $x = \mathrm{norm}([x_1(1), x_2(1)])$;
   if $x == 0$
      $\alpha = a * [1, 0]$;
   else
      $\alpha = -(a/x) * ([x_1(1), x_2(1)])$;
   end
   $u_1(1, 1:2) = u_1(1, 1:2) - \alpha$; u =Q_Comp($u_1(:,1), u_1(:,2)$); $\beta = 1/(a*(a+x))$;
end

Function: G=QC_Givens(g_1, g_2)
   $s = \sqrt{g_1 * \overline{g_1} + g_2 * \overline{g_2}}$; $tol = 1e-14$;
   if $s <= tol$
      $G = \mathrm{eye}(2)$;
   else

$$G = \begin{bmatrix} g_1 & g_2 \\ -\overline{g_2} & \overline{g_1} \end{bmatrix} /s \;;$$

   end
end

Next, we test the performance and characteristics of the above algorithm (CQQRD) by numerical experiments.

Example 3.1 For the quaternion matrix $A = A_1 + A_2\mathrm{i} + A_3\mathrm{j} + A_4\mathrm{k} \in \mathbf{H}^{m\times n}$ generated by matlab's function rand, where $A_1 = \mathrm{rand}(m, n), A_2 = \mathrm{rand}(m, n), A_3 = \mathrm{rand}(m, n), A_4 = \mathrm{rand}(m, n)$, $m = 10k, n = 8k$, for $k = 1 : 50$, perform six different methods to calculate the QR decomposition of A: the complex structure-preserving algorithm CQQRD, the function qr of Quaternion Toolbox [17], the real structure-preserving algorithm qQR1, qQR2, qQR3, qQR4 [18]. Test the CPU times and computational errors of the above algorithms.

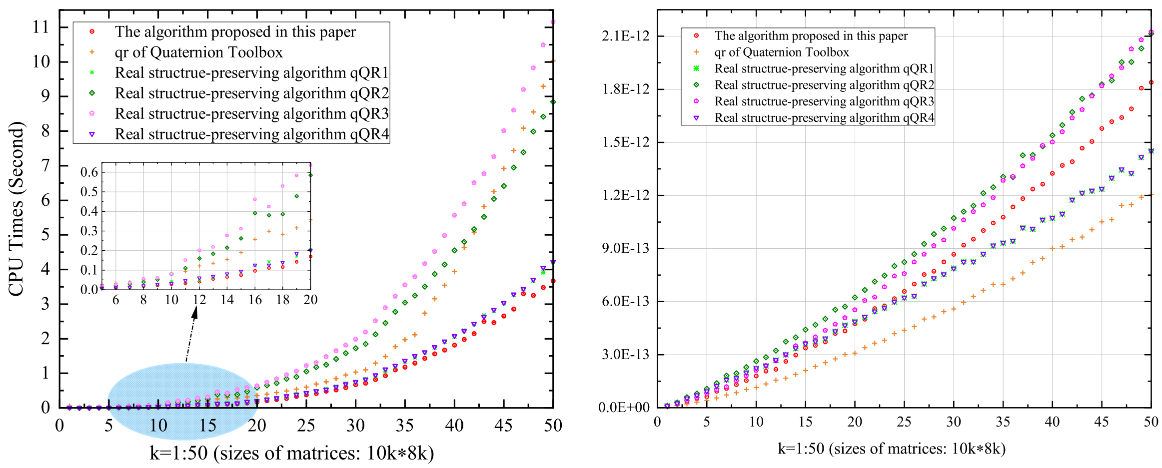


Figure 1: CPU times and Errors for computing QR decomposition of A

From Fig. 1, it can be easily find that our proposed complex structure-preserving algorithm CQQRD requires less CPU time for QR decomposition of quaternion matrices, i.e., the computational efficiency of the proposed methods in this paper is better than the rest of the algorithms. Moreover, all the above algorithms are able to achieve smaller computational errors. Therefore, the following experiments of color image watermarking algorithm will be conducted based on CQQRD.

4 Blind watermarking algorithm for color images based on CQQRD

Most QR decomposition-based watermarking algorithms convert each pixel point of a color watermark image into an 8-bit binary number, but it leads to a small embedding capacity of the host image. In this section, a new watermarking algorithm for color images based on CQQRD is proposed, we will embed by 6-bit trinary method, for example, each pixel point in each channel of a color image is recorded as an integer no greater than 255 on the computer, then 255 converted to trinary can be recorded as '100110'. This method can effectively increase the information hiding capacity of the host image and improve the information hiding concealment. It is well known that the watermarking algorithm mainly consists of two stages, embedding the watermark image into the host image and extracting the watermark image. In the following, we will introduce the algorithm steps of the corresponding phases.

4.1 Color watermarking algorithm embedding strategy

Step 1. Separate the watermark image $W = W_1\mathrm{i} + W_2\mathrm{j} + W_3\mathrm{k} \in \mathbf{H}^{s\times t}$ and the host image $H = H_1\mathrm{i} + H_2\mathrm{j} + H_3\mathrm{k} \in \mathbf{H}^{m\times n}$ into two-dimensional images $W_x \in \mathbf{R}^{s\times t}$ and $H_x \in \mathbf{R}^{m\times n}$, $x = 1, 2, 3$ for the red, green and blue channels, respectively. To improve the attack resistance of the watermarking algorithm, perform Arnold transform (2.9) on W_x separately, i.e., add the private key PK_1. Then convert W_x into 6-bit trinary form to obtain vectors $\widetilde{W}_x \in \mathbf{R}^{1\times 6st}$, where each element $\widetilde{W}_x(z)$ of $\widetilde{W}_x$ belongs to $\{0, 1, 2\}$, and $z \in \{1, 2, \cdots, 6st\}$.

Step 2. Partition the host image H into many small non-overlapping 3×3 block matrices $H_{\{u,v\}}$, where $u \in \{1, 2, \cdots, \text{floor}(m/3)\}$, $v \in \{1, 2, \cdots, \text{floor}(n/3)\}$ represent the coordinate positions of the small matrix $H_{\{u,v\}}$ in H, and randomly select the $6st$ small matrices to be used as the embedding targets based on Equations (2.10) and (2.11), i.e., add the private key PK_2. Perform QR decomposition of the selected small matrices $H_{\{u,v\}}$ by the algorithm CQQRD, which is easily obtained in the following form,

$$H_{\{u,v\}} = Q_{\{u,v\}} \times R_{\{u,v\}}, \tag{4.1}$$

where $Q_{\{u,v\}}, R_{\{u,v\}} \in \mathbf{H}^{3\times 3}$ are the unitary quaternion matrix and the upper triangular quaternion matrix, respectively.

Step 3. Take the two elements $(q_{21})_x$ and $(q_{31})_x$ of the second and third rows of the first column of the three coefficient matrix $(Q_{\{u,v\}})_x$ of the three imaginary parts $\mathrm{i}, \mathrm{j}, \mathrm{k}$ of $Q_{\{u,v\}}$. Then calculate the new $(q_{21})'_x$ and $(q_{31})'_x$ based on the following rules,

$$\text{if } \widetilde{W}_x(z) = 0 \ \& \ |q_{21} - q_{31}|_x > T, \quad \text{then} \begin{cases} (q_{21})'_x = (q_{21} + q_{31})_x/2 \\ (q_{31})'_x = (q_{21} + q_{31})_x/2 \end{cases}, \tag{4.2a}$$

$$\text{if } \widetilde{W}_x(z) = 1 \ \& \ (q_{21} - q_{31})_x \le T, \quad \text{then} \begin{cases} (q_{21})'_x = (q_{21} + q_{31})_x/2 + T \\ (q_{31})'_x = (q_{21} + q_{31})_x/2 - T \end{cases}, \tag{4.2b}$$

$$\text{if } \widetilde{W}_x(z) = 2 \ \& \ (q_{21} - q_{31})_x > -T, \quad \text{then} \begin{cases} (q_{21})'_x = (q_{21} + q_{31})_x/2 - T \\ (q_{31})'_x = (q_{21} + q_{31})_x/2 + T \end{cases}, \tag{4.2c}$$

where T is the threshold of embedding watermark.

Step 4. Using $(q_{21})'_x$ and $(q_{31})'_x$ to replace $(q_{21})_x$ and $(q_{31})_x$ to obtain $Q'_{\{u,v\}}$, and then it is easy to obtain $H'_{\{u,v\}}$ by the matrix multiplication below,

$$H'_{\{u,v\}} = Q'_{\{u,v\}} \times R_{\{u,v\}}, \tag{4.3}$$

until all the elements in $\widetilde{W}_x$ are used, we can get a new host image H' with an embedded watermark image. It is worth noting that the quaternion matrix H' obtained by this algorithm is still an approximately pure imaginary quaternion matrix, i.e., the real part of H' is a matrix that is approximately equal to zero.

4.2 Color watermarking algorithm extraction strategy

Step 1. Partition the new host image $H' = H'_1\mathrm{i} + H'_2\mathrm{j} + H'_3\mathrm{k} \in \mathbf{H}^{m\times n}$ containing the watermark image into many non-overlapping 3×3 matrices.

Step 2. Select $6st$ small matrices $H'_{\{u,v\}} \in \mathbf{H}^{3\times 3}$ with embedded watermarked image information according to the private key PK_2 order.

Step 3. Perform QR decomposition of $H'_{\{u,v\}}$ based on CQQRD algorithm. The form is as follows:

$$H'_{\{u,v\}} = Q'_{\{u,v\}} \times R'_{\{u,v\}}, \tag{4.4}$$

where $Q'_{\{u,v\}}, R'_{\{u,v\}} \in \mathbf{H}^{3\times 3}$ are the unitary quaternion matrix and the upper triangular quaternion matrix, respectively.

Step 4. Take the two elements $(q_{21})'_x$ and $(q_{31})'_x$ of the second and third rows of the first column of the three coefficient matrix $(Q_{\{u,v\}})'_x$ of the three imaginary parts $\mathrm{i}, \mathrm{j}, \mathrm{k}$ of $Q'_{\{u,v\}}$. Then compute the corresponding trinary sequence $\widetilde{W'_x} \in \mathbf{R}^{1\times 6st}$ of the watermark image based on the following rules,

$$\widetilde{W'_x}(z) = \begin{cases} 1, & \text{if } (q'_{21} - q'_{31})_x > T \\ 2, & \text{elseif } (q'_{21} - q'_{31})_x \leq -T \\ 0, & \text{else} \end{cases}. \tag{4.5}$$

Step 5. Convert $\widetilde{W'_x}$ into a decimal vector, then collapse it into a matrix of $s \times t$ by rows, finally obtain W'_x based on the private key PK_1 inverse Arnold transform. Let $W' = W'_1\mathrm{i} + W'_2\mathrm{j} + W'_3\mathrm{k} \in \mathbf{H}^{s\times t}$, then W' is the extracted watermark image.

Remark. Watermark capacity is also an important factor in measuring watermarking algorithms that should not be ignored. The watermark capacity of an image depends on various factors, such as the size and the resolution of the image, watermark embedding algorithm, etc. For the color watermarking algorithm in this paper, the maximum embedding capacity of a 512×512 color host image is $170 \times 170 \times 3$ bits. Therefore, in order to ensure the invisibility of the algorithm and to facilitate comparison with other methods, we also use a 32×32 color watermark image for embedding, it is easy to calculate the embedding ratio of the proposed algorithm is $(32 \times 32 \times 6 \times 3)/(512 \times 512 \times 3) \approx 0.0234$. Generally, if the embedding ratio of the watermark is smaller, the invisibility of the host image is better.

5 Experimental analysis

In this section, we will test the performance as well as the robustness of the color watermarking algorithm in this paper through specific numerical experiments.

Example 5.1 Given four original 512×512 color host images Lenna, Airplane, House, Peppers and one original 32×32 color watermark image, observe the relationship between the watermarked host image and the extracted watermark image with the threshold T by the watermark embedding and extracting algorithm in Section 4, where $T = 0.015 : 0.005 : 0.1$. Moreover, when $T = 0.08$, show the watermarked host image and the extracted watermark images.

In general, we use the PSNR and NC values to describe the relationship between the watermarked host image and the extracted watermark image with the original

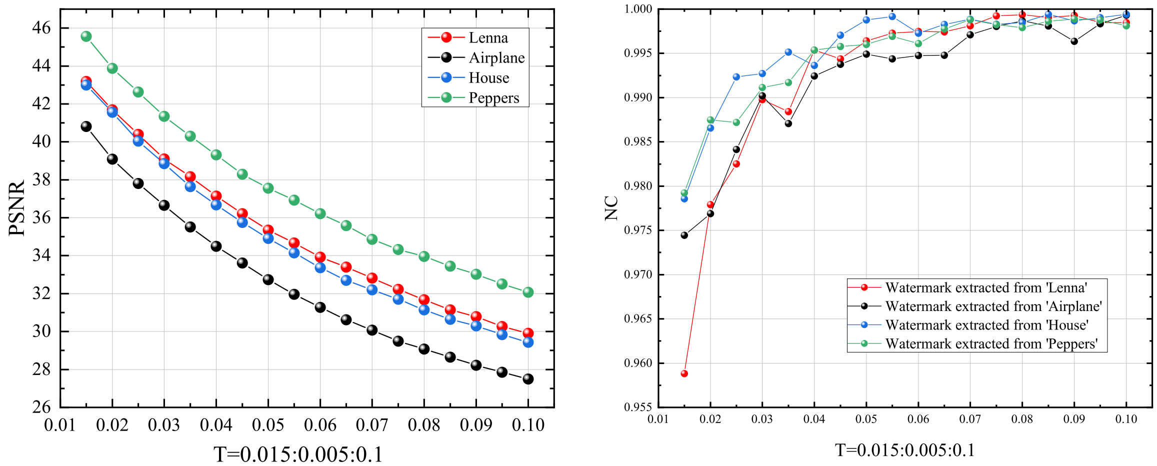
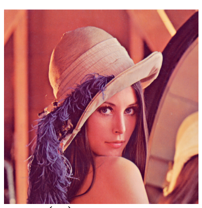


Figure 2: PSNR values of the watermarked host images and NC values of the extracted watermark images


(a) Lenna


(b) Watermark


(c) PSNR=31.6867


(d) NC=0.9992


(a) Airplane

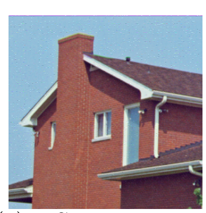

(b) Watermark


(c) PSNR=29.0520


(d) NC=0.9988


(a) House


(b) Watermark


(c) PSNR=31.1921


(d) NC=0.9992

images, respectively. From the Fig. 2 above, it can be easily find that the PSNR value of the watermarked host image decreases as the threshold T increases, and the NC value of the extracted watermark tends to increase roughly with the increase of the threshold T, gradually approaching to 1. In order to preserve the invisibility of the watermarking algorithm and to consider the effect of extracting the watermark, we choose $T = 0.08$, and Fig. 3 shows the results of the watermarking algorithm performed for the four color host images, where (a) and (b) denote the original color host image and the watermark image, respectively, (c) denotes the watermarked host

(a) Peppers (b) Watermark (c) PSNR=33.9282 (d) NC=0.9998

Figure 3: The results regarding the watermarking algorithm are executed on four host images at T=0.08

image, and (d) denotes the extracted watermark image. This experiment proves that our proposed watermarking method is effective in the absence of attacks.

Another important metric of the watermarking algorithm is its robustness, i.e., its ability to resist attacks such as noise. Next we test some common noises and compare them with several watermarking algorithms to analyze the performance as well as the advantages and disadvantages of watermarking algorithms based on quaternion complex structure-preserving QR decomposition.

Example 5.2 Select the 512×512 color Lenna image and the 32×32 color image in Example 5.1 as the host image and the watermark image, respectively. In the absence of attacks, test the trends of PSNR values of the watermarked host image and NC values of the extracted watermark image with variables T and ρ_1, ρ_2 by three methods, respectively, which are the proposed method and the methods in the paper [13] and [19], respectively. Then choose a suitable variable from each method to test the impact of several common attacks on the three algorithms.

Proposed method			Zhang [13]			Li [19]		
Threshold T	PSNR	NC	Scale factor ρ_1	PSNR	NC	Scale factor ρ_2	PSNR	NC
0.02	41.7553	0.9749	0.03	61.1530	0.9481	0.07	49.8741	0.9992
0.05	35.4691	0.9977	0.12	45.7441	0.9947	0.14	43.1897	0.9993
0.08	31.6267	0.9996	0.21	40.4358	0.9948	0.21	39.4169	0.9991
0.12	28.3935	1.0000	0.3	37.1401	0.9948	0.28	36.7952	0.9989

Table 1. PSNR values of the watermarked host image and NC values of the extracted watermark image

Attack types	Proposed method	Zhang [13]	Li [19]
Salt & pepper noise (0.02)	0.9800	0.8932	0.9022
Salt & pepper noise (0.05)	0.9716	0.7783	0.8240
Gaussian noise (0.1)	0.9123	0.6100	0.7389
Gaussian noise (0.5)	0.9113	0.5153	0.6708
Speckle noise (0.05)	0.8760	0.8154	0.9024
Adjust brightness [0,0.6]	0.8259	0.9544	0.9798

Table 2. The effects of several attacks on the three algorithms (NC values)

From Table 1 above, it can be easily find that the invisibility of papers [13] and [19] is better when the difference between the selected host image and the watermark image is large, but the proposed method in this paper is also able to obtain better results, and in addition, the proposed method is able to obtain better results for extracted watermark image. In order to ensure the invisibility of each algorithm and the better effect of extracted watermark images, we choose $T = 0.08, \rho_1 = 0.21, \rho_2 = 0.14$, and test the effect for several common attacks as shown in Table 2, which obviously shows that the proposed method in this paper has strong resistance to salt & pepper noise. Moreover, the resistance of the proposed algorithm for larger Gaussian noise is also better than the other two methods, and for Speckle noise, the anti-attack effect of all three methods is less satisfactory. For the adjustment of image brightness, the other two methods are better than the method proposed in this paper.

6 Conclusions

This paper, by means of a complex representation form of quaternion matrices, proposes a complex structure-preserving algorithm for QR decomposition of quaternion matrices, which not only outperforms existing methods in terms of computational efficiency, but also the diagonal elements of the R-matrices obtained after decomposition are real numbers. This paper also applies the proposed complex structure-preserving QR decomposition method to color watermarking algorithm, in which uses 6-bit trinary instead of the traditional 8-bit binary in order to improve the watermark embedding capacity of the algorithm. Moreover, this paper uses two private keys for encryption in order to improve the security of the algorithm, the experiments prove that the watermarking algorithm in this paper has better performance and robustness to salt and pepper noise, etc.

Acknowledgement

The section 2 is supported by the Ministry of Science and Higher Education of the Russian Federation, supplementary agreement (No. 075-02-2023-947, February 16, 2023) and the sections 3-6 are supported by the Chinese Government Scholarship (CSC No. 202108370087).

References

[1] Barni M., Bartolini F., Piva A., Multichannel watermarking of color images, IEEE Transactions on circuits and systems for video technology, 12.3 (2002), 142-156.

[2] Tsui T., Zhang X., Androutsos D., Color image watermarking using multidimensional Fourier transforms, IEEE Transactions on Information Forensics and security, 13.1 (2008), 16-28.

[3] Abraham J., Paul V., An imperceptible spatial domain color image watermarking scheme, Journal of King Saud University-Computer and Information Sciences, 31.1 (2019), 125-133.

[4] Chou C. and Wu T., Embedding color watermarks in color images, EURASIP Journal on Advances in Signal Processing, 2003, 1-9.

[5] Abdulrahman A. and Ozturk S., A novel hybrid DCT and DWT based robust watermarking algorithm for color images, Multimedia Tools and Applications, 78 (2019), 17027-17049.

[6] Liu D., Su Q., Yuan Z. and Zhang X., A blind color digital image watermarking method based on image correction and eigenvalue decomposition, Signal Processing: Image Communication, 95 (2021), 116292.

[7] Sun Y., Su Q., Chen S. and Zhang X., A double-color image watermarking algorithm based on quaternion Schur decomposition, Optik, 269 (2022), 169899.

[8] Hamilton W., On a new species of imaginary quantities, connected with the theory of quaternions, Proceedings of the Royal Irish Academy, 2 (1840), 424-434.

[9] Pei S. and Cheng C., A novel block truncation coding of color images using a quaternion-moment-preserving principle. IEEE Transactions on Communications, 45.5 (1997), 583-595.

[10] Bas P., Le Bihan N. and Chassery J., Color image watermarking using quaternion Fourier transform, 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). IEEE, 3 (2003), III-521.

[11] Wang X., Liu Y., Han M. and Yang H., Local quaternion PHT based robust color image watermarking algorithm, Journal of Visual Communication and Image Representation, 38 (2016), 678-694.

[12] Chen Y., Jia Z., Peng Y. and Zhang D., A new structure-preserving quaternion QR decomposition method for color image blind watermarking, Signal Processing, 185 (2021), 108088.

[13] Zhang M., Ding W., Li Y., Sun J. and Liu Z., Color image watermarking based on a fast structure-preserving algorithm of quaternion singular value decomposition, Signal Processing, 208 (2023), 108971.

[14] Jiang T., An algorithm for eigenvalues and eigenvectors of quaternion matrices in quaternionic quantum mechanics, Journal of mathematical physics, 45.8 (2004), 3334-3338.

[15] Peterson G., Arnold's cat map, Math Linear Algebra, 45 (1999), 1-7.

[16] Wei M., Li Y., Zhang F. and Zhao J., Quaternion matrix computations, Nova Science Publishers, Incorporated, New York, 2018.

[17] Sangwine S., Le Bihan N., Quaternion toolbox for matlab, http://qtfm.sourceforge.net/.

[18] Li Y., Wei M., Zhang F., Zhao J., Real structure-preserving algorithms of Householder based transformations for quaternion matrices, Journal of Computational and Applied Mathematics, 305 (2016), 82-91.

[19] Li Y., Wei M., Zhang F., Zhao J., A new double color image watermarking algorithm based on the SVD and Arnold scrambling, Journal of Applied Mathematics, 2016.

Zhenwei Guo,
North-Eastern Federal University,
58 Belinsky str, Yakutsk, Republic of Sakha (Yakutia), Russia, 677000,
Email: `guozhenweilcu@163.com`.

Received 18.09.2023, revised 04.10,2023, Accepted 06.10.2023